NX 奈学教育

Kafka消息引擎底层架构深度剖析

主讲人：李希沅

2020.09.08

课程目标

NX 奈学教育

- 了解Kafka的特性
- 揭秘Kafka Producer高性能架构设计方案
- 深度剖析Kafka服务端高并发，高性能，高可用的架构设计
- 深度剖析Kafka提升Consumer的稳定性设计
- 让大家深入了解Kafka的设计，合理对Kafka进行调优

课程目录

奈学教育

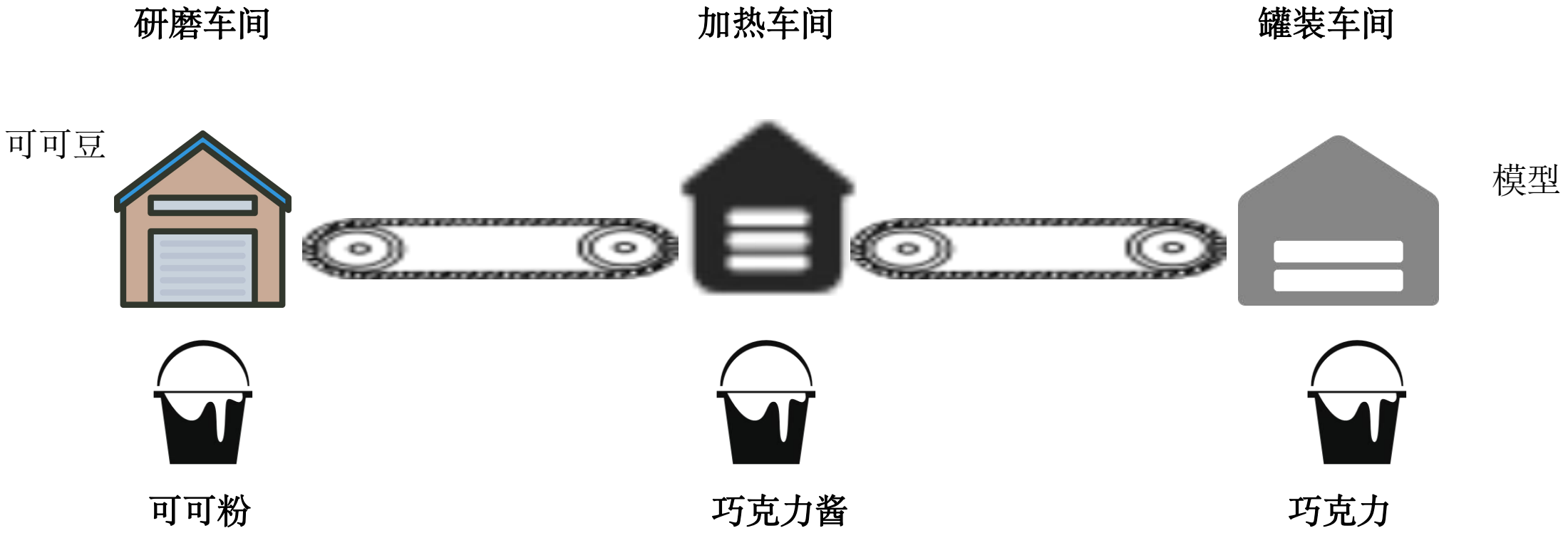
第一章Kafka核心概念介绍

第二章Kafka高性能架构鉴赏

第一章：Kafka核心概念介绍

为什么会有消息系统

NX 奈学教育



问题：每道工序的生产速度不同，就会导致上下游的工人相互等待！

为什么会有消息系统

研磨车间　　加热车间　　罐装车间

仓库　　仓库

可可粉　　巧克力酱　　巧克力

仓库的作用就类似于消息系统对于应用程序的作用

NX 奈学教育

常见的消息系统

1. RabbitMQ
2. ActiveMQ
3. Kafka

.....

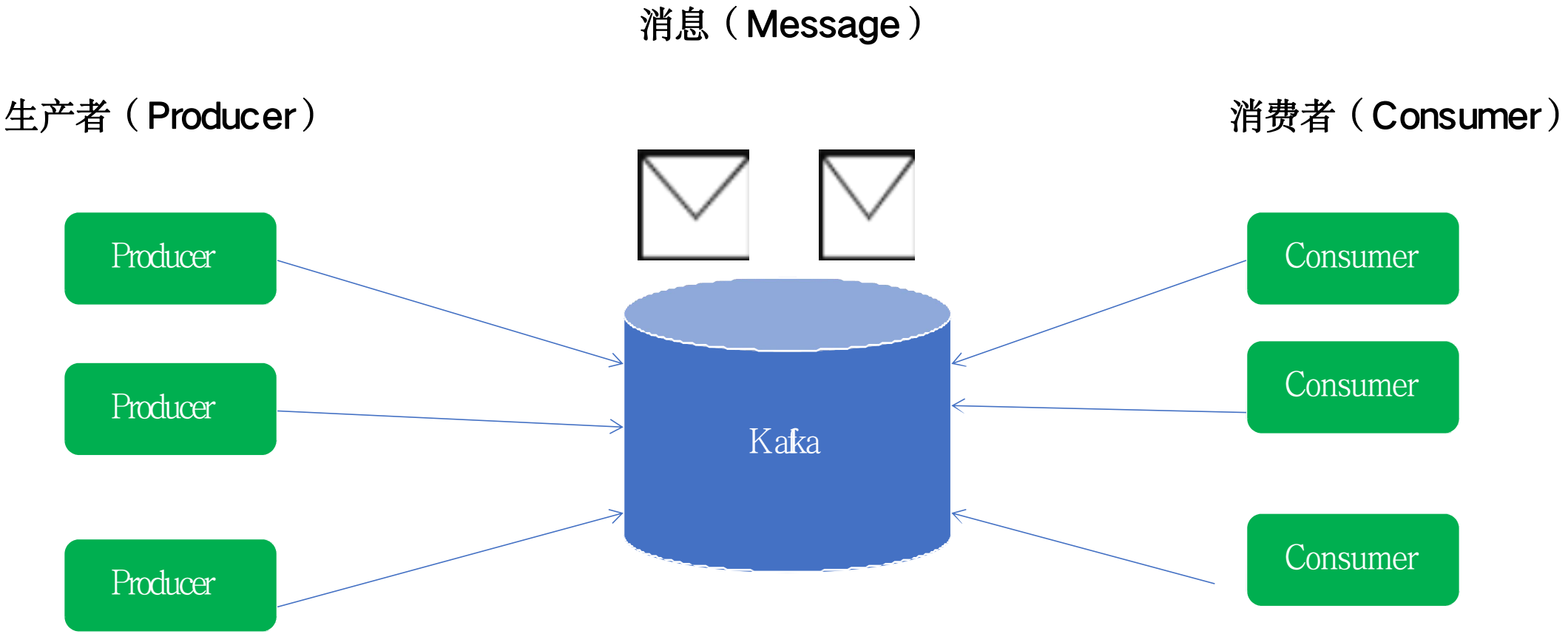
为什么学习Kafka?

1. 互联网最火的技术：A（AI）B（BigData）C(Cloud)，B最容易落地
2. 大数据里经常需要应对数据激增，数据复杂度增加，数据变化速率变快等场景
3. Kafka能轻松解决这些问题

大数据领域的消息系统-Kafka

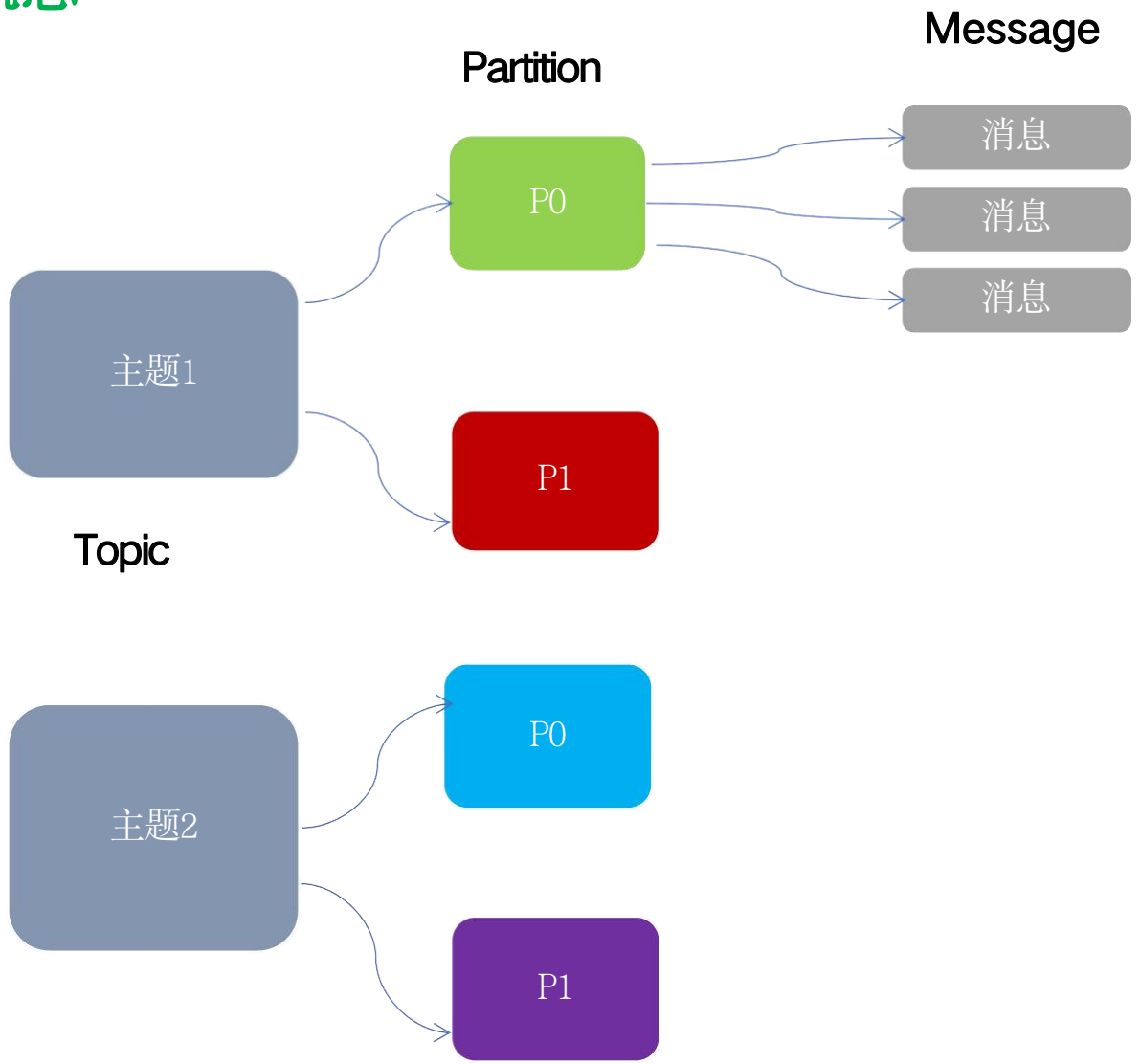
NX 奈学教育

Kafka是一个开源的高吞吐量的分布式发布订阅消息系统

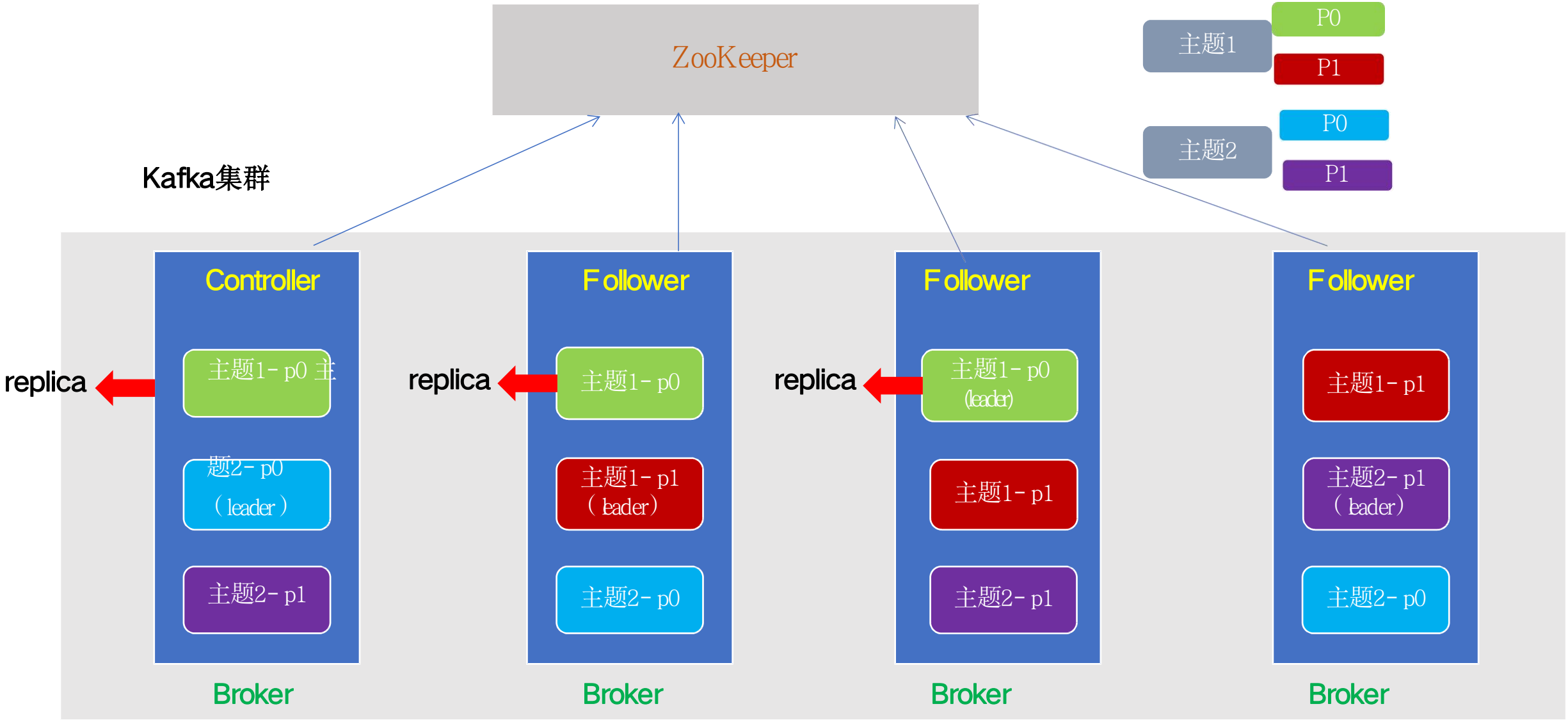


Kafka核心概念

NX 奈学教育



Kafka核心概念


ZooKeeper
主题1
P0
P1
主题2
P0
P1
Kafka集群
Controller
Follower
Follower
Follower
replica
replica
replica
主题1- p0 主
题2- p0
（leader）
主题2- p1
主题1- p0
主题1- p1
（leader）
主题2- p0
主题1- p0
(leader)
主题1- p1
主题2- p1
主题1- p1
主题2- p1
（leader）
主题2- p0
Broker
Broker
Broker
Broker

核心概念总结

NX 奈学教育

Producer（生产者）：发送消息

Consumer（消费者）：订阅消息

Broker：Kafka的节点

Controller：Kafka服务器的主节点

Follower：Kafka服务的从节点

Topic：主题，类似于数据库里的表

Partition：分区，一个主题可以有多个分区，类似于数据库里的分区

Replica:副本，为了保证数据安全，每个Partition可以设置多个副本（leader replica，Follower replica）

Message：消息，消息存在分区里

Offset：消息存储进度/消费者的消费进度

第二章 Kafka高性能架构鉴赏

思考

NX 奈学教育

Kafka是一个把性能用到极致的框架，是一个支持高并发，高性能，高可用的分布式消息系统，假如是你，你怎么去设计他的服务端呢？你会从哪些角度去提升呢？

网络
磁盘
内存
CPU

服务端设计-服务端请求是如何处理的？

NX 奈学教育

想法一：顺序处理请求

```
while（true）{
        Request request=accept(connection); handle(request);
}
```

每个请求必须等待前一个请求处理完了以后才会得到处理，吞吐量太差！

服务端设计-服务端请求是如何处理的？

NX 奈学教育

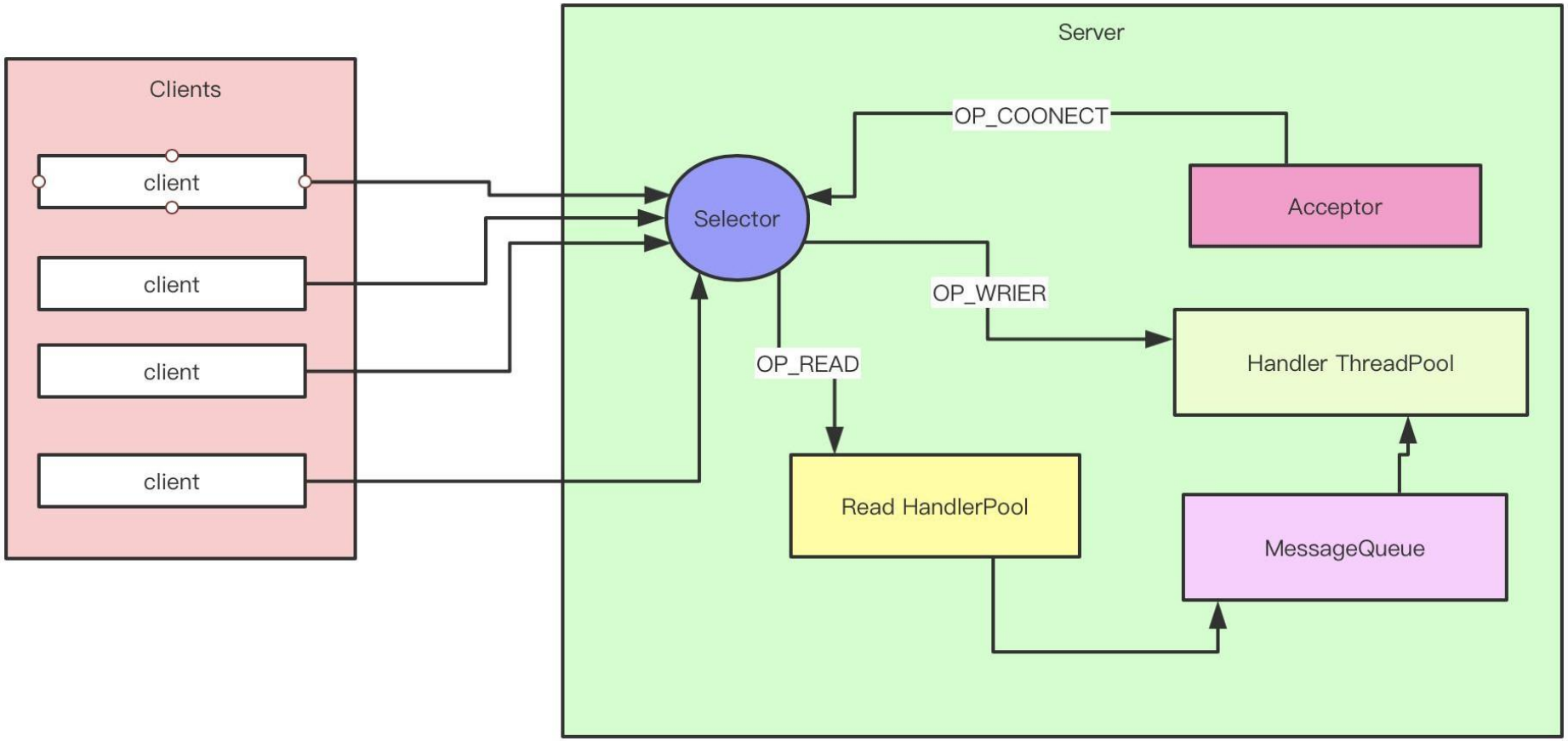
想法二：异步处理

```
while（true）{
        Request request=  accept(connection);
        Thread    thread=  new Thread（handle(request)）；  thread.start();
        }
```

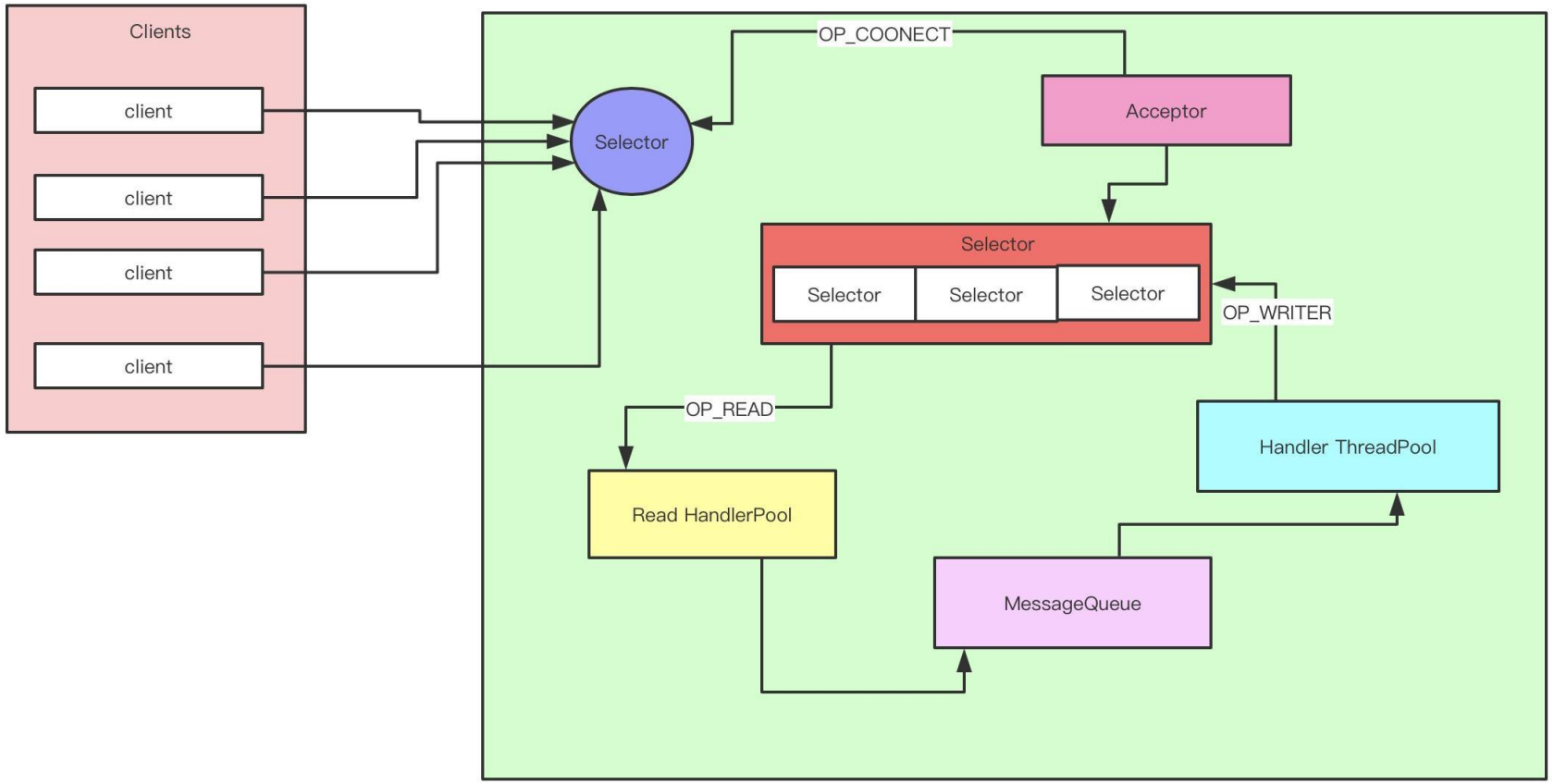
每个请求都创建一个线程去处理，请求不会阻塞，但是每个请求都创建一个线程开销太大

服务端设计-服务端请求是如何处理的？

NX 奈学教育

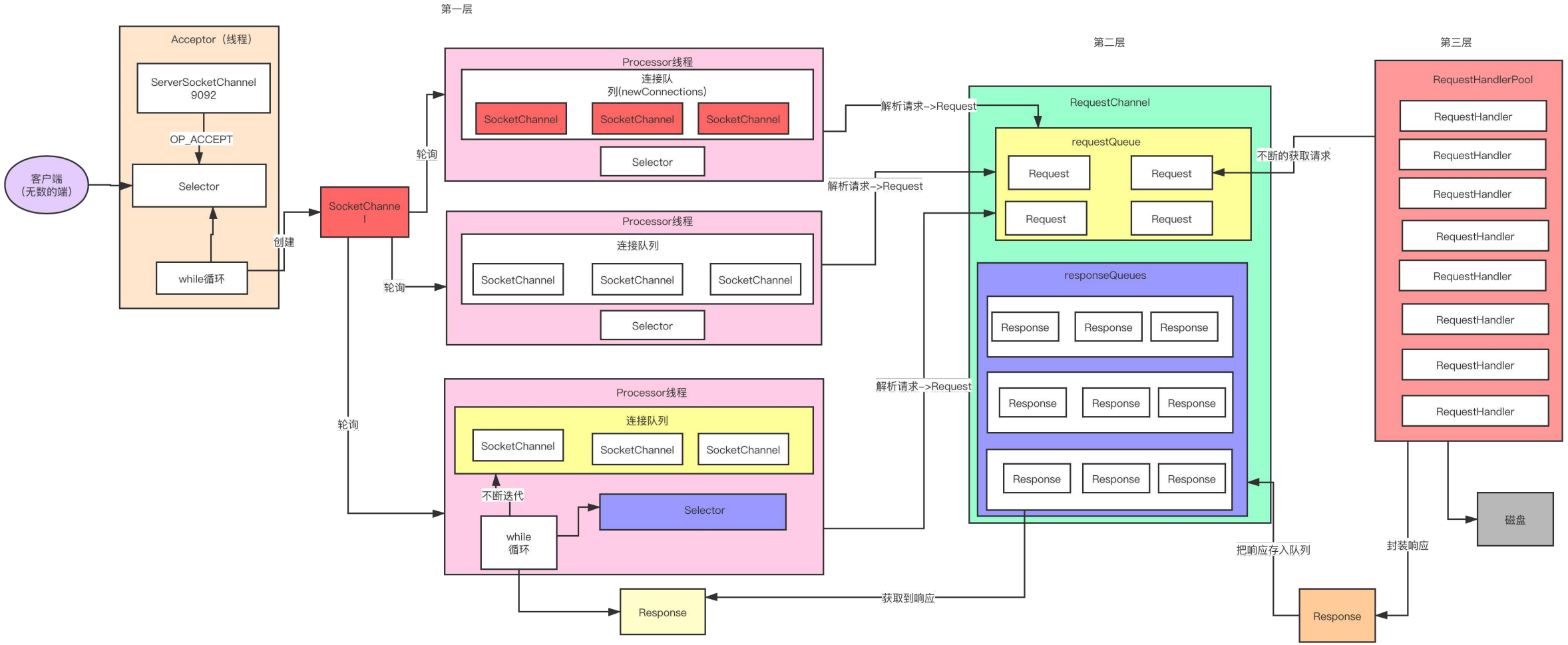


服务端设计-服务端请求是如何处理的？Reactor设计模式



服务端设计-服务端请求是如何处理的-高性能-高并发

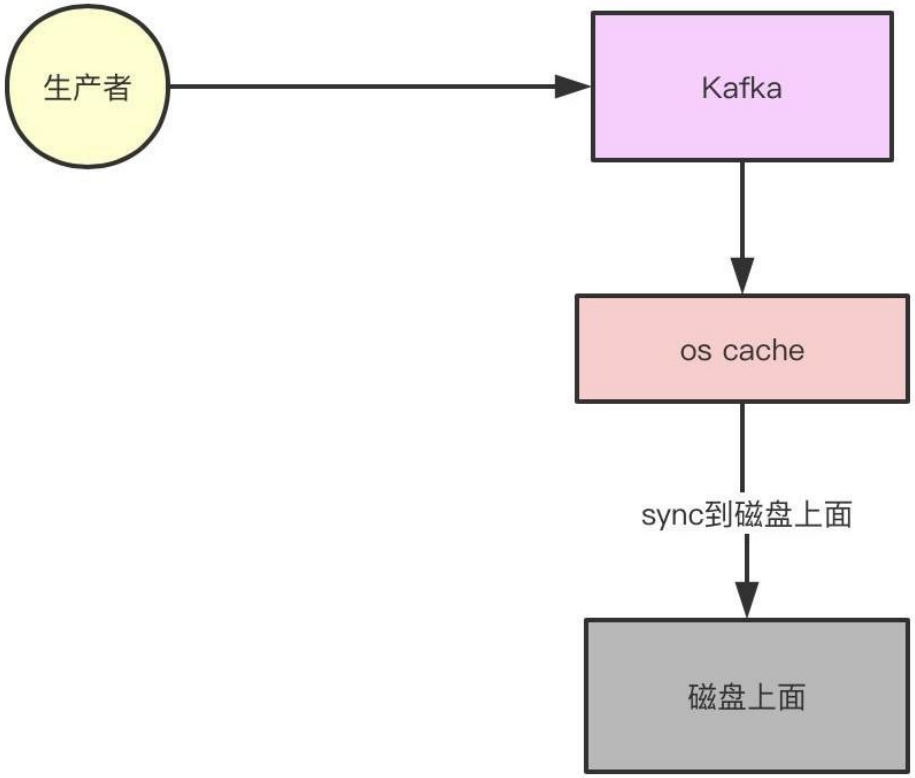
num.io.threads=8



num.network.threads=3

服务端设计-顺序读写-高性能

NX 奈学教育

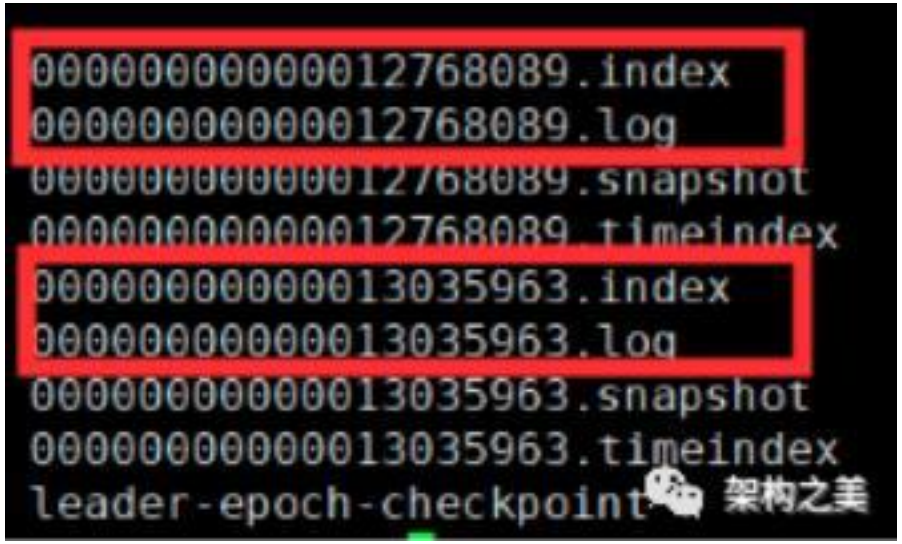


Kafka是将消息记录持久化到本地磁盘中的，一般人会认为磁盘读写性能差，对Kafka性 能如何保证提出质疑。

实际上不管是内存还是磁盘，快或慢关键在于寻址的方式，磁盘分为顺序读写与随机读写，内存也一样分为顺序读写与随机读写。基于磁盘的随机读写确实很慢，但磁盘的顺序读写性能却很高，一般而言要高出磁盘随机读写三个数量级，一些情况下磁盘顺序读写性能甚至要高于内存随机读写

服务端设计-跳表设计

NX 奈学教育

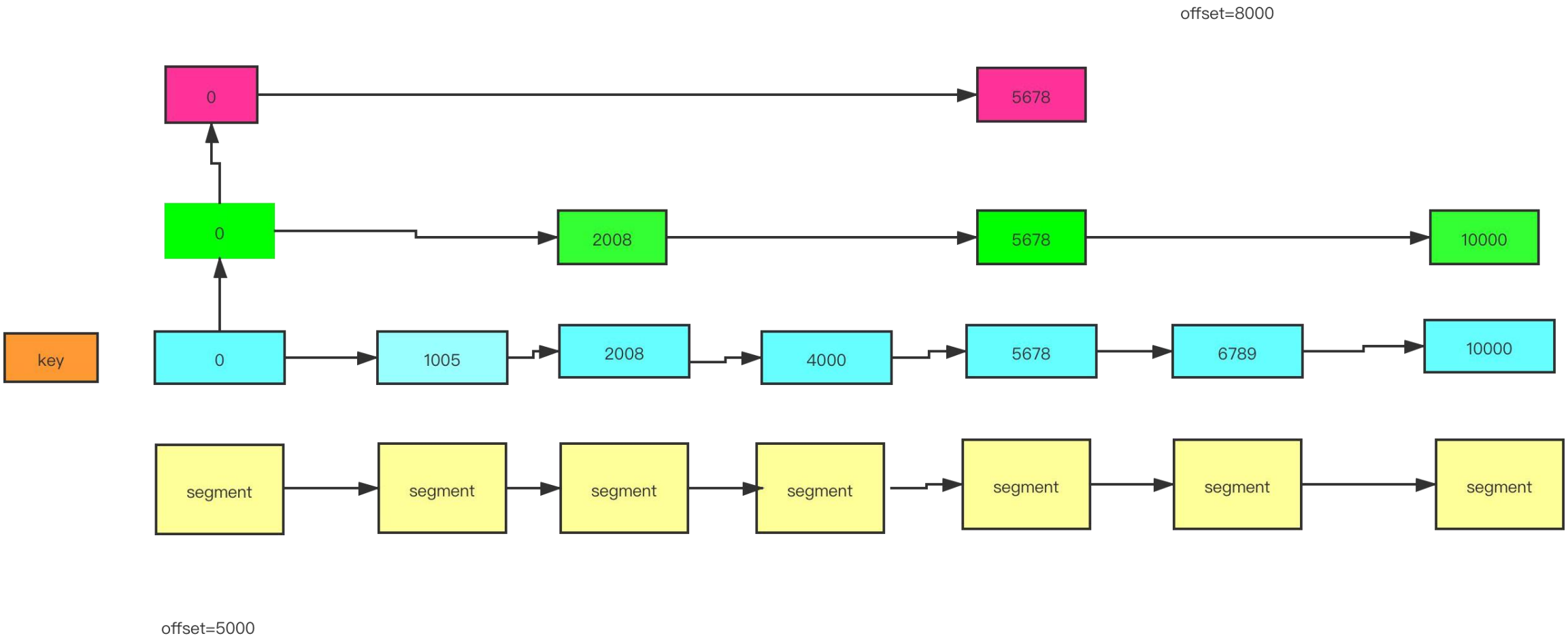


log文件：里面存储的是消息 index：存储索引信息

这两个文件的文件名都是一样的，成对出现的，这个文件名是以log文件里的第一条消息的offset命名的，如下第一个文 件的文件名叫00000000000012768089，代表着这个文件 里的第一个消息的offset是12768089，也就是说第二条消 息就是12768090了。

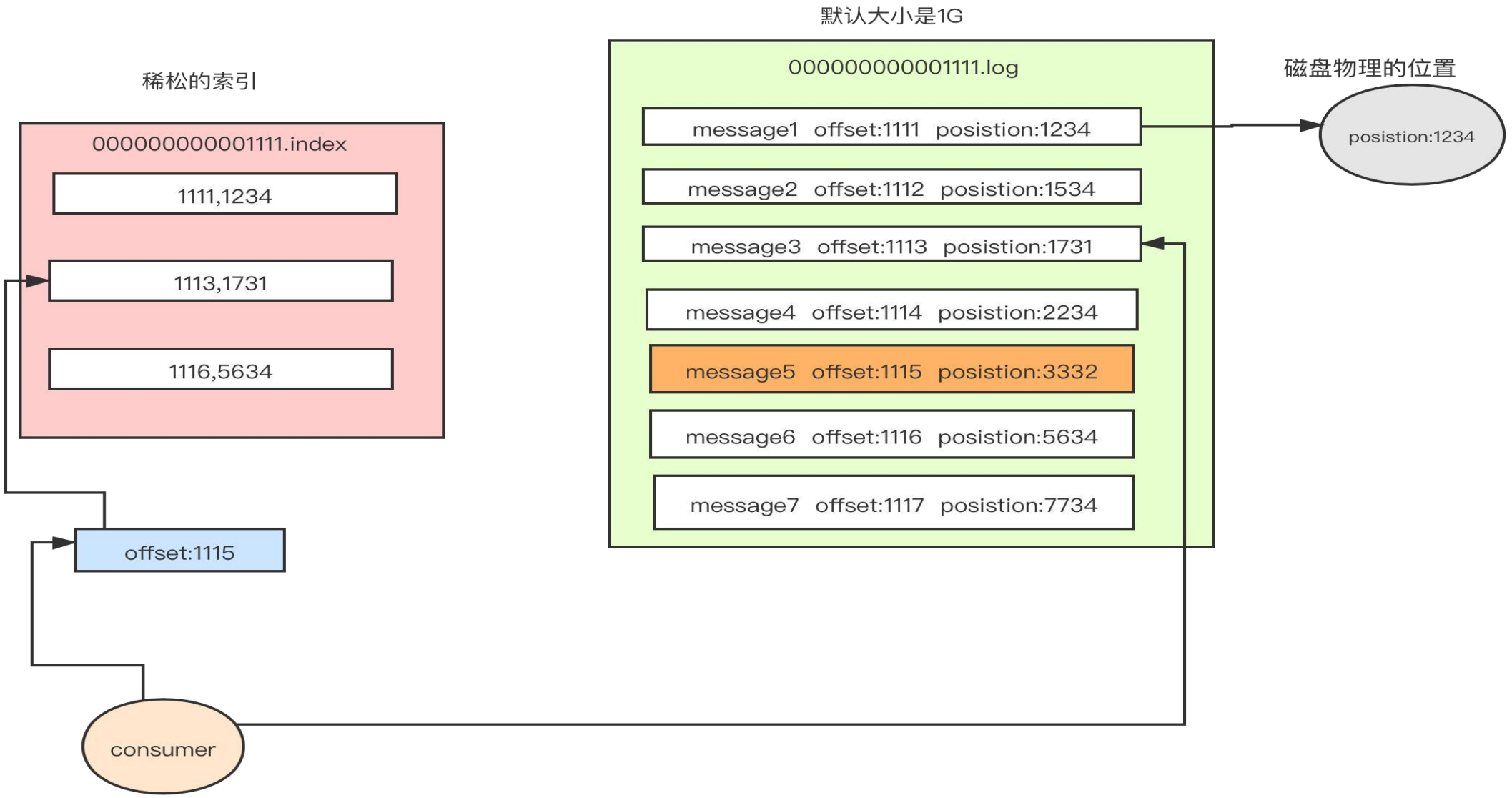
服务端设计-跳表设计

NX 奈学教育



服务端设计-稀松索引

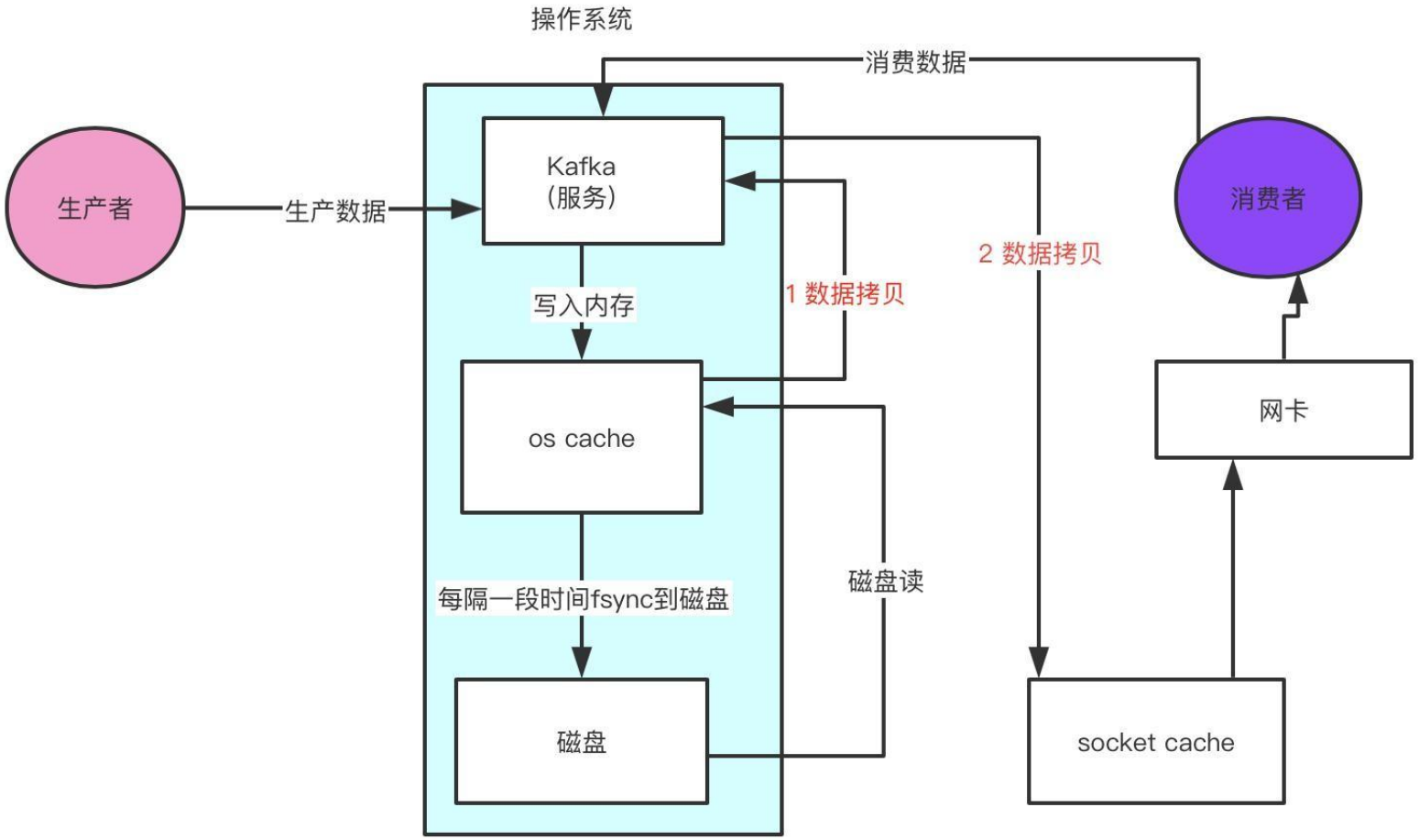
NX 奈学教育



服务端设计-零拷贝

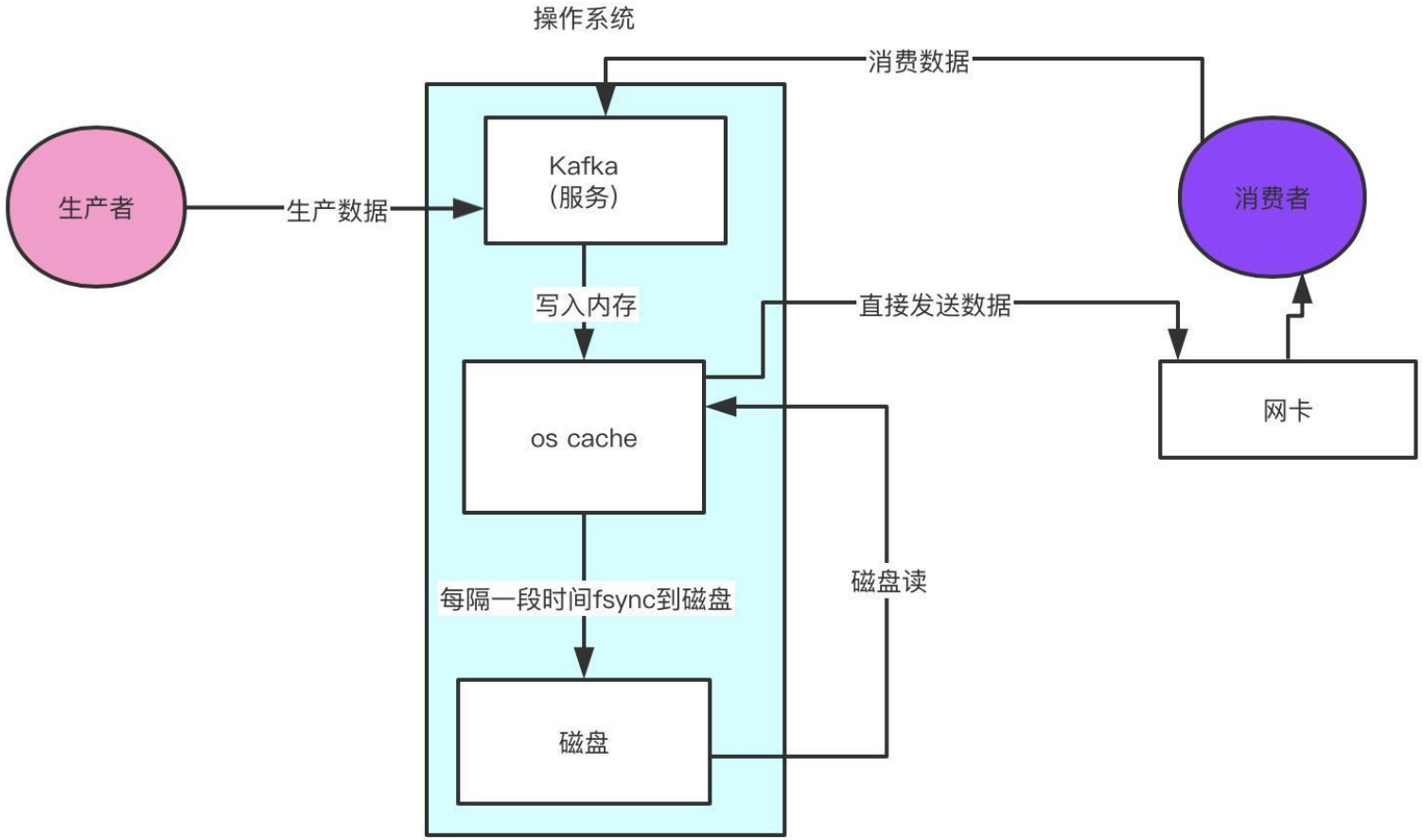
NX 奈学教育

非零拷贝



服务端设计-零拷贝

NX 奈学教育



服务端设计-总结

NX 奈学教育

高并发，高性能的网络设计

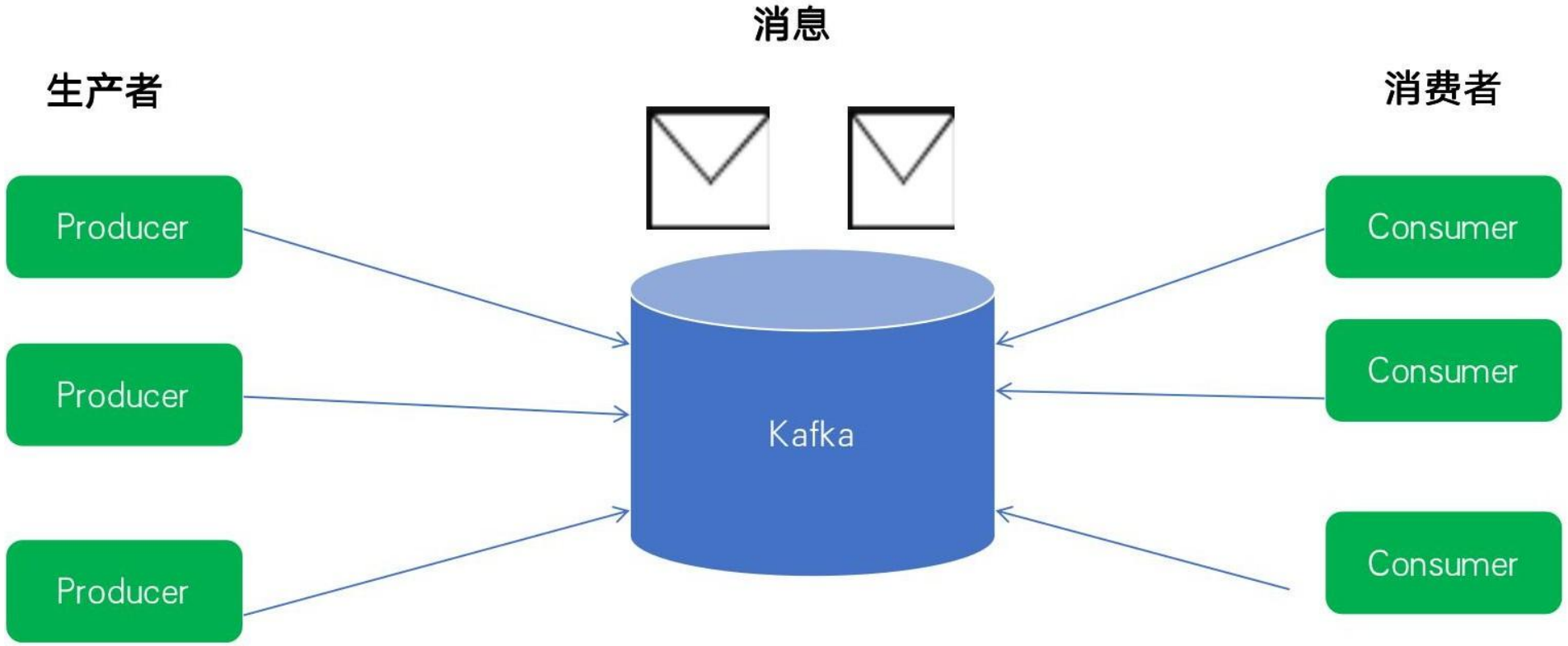
顺序读写

跳表设计 稀松索引 零拷贝

思考

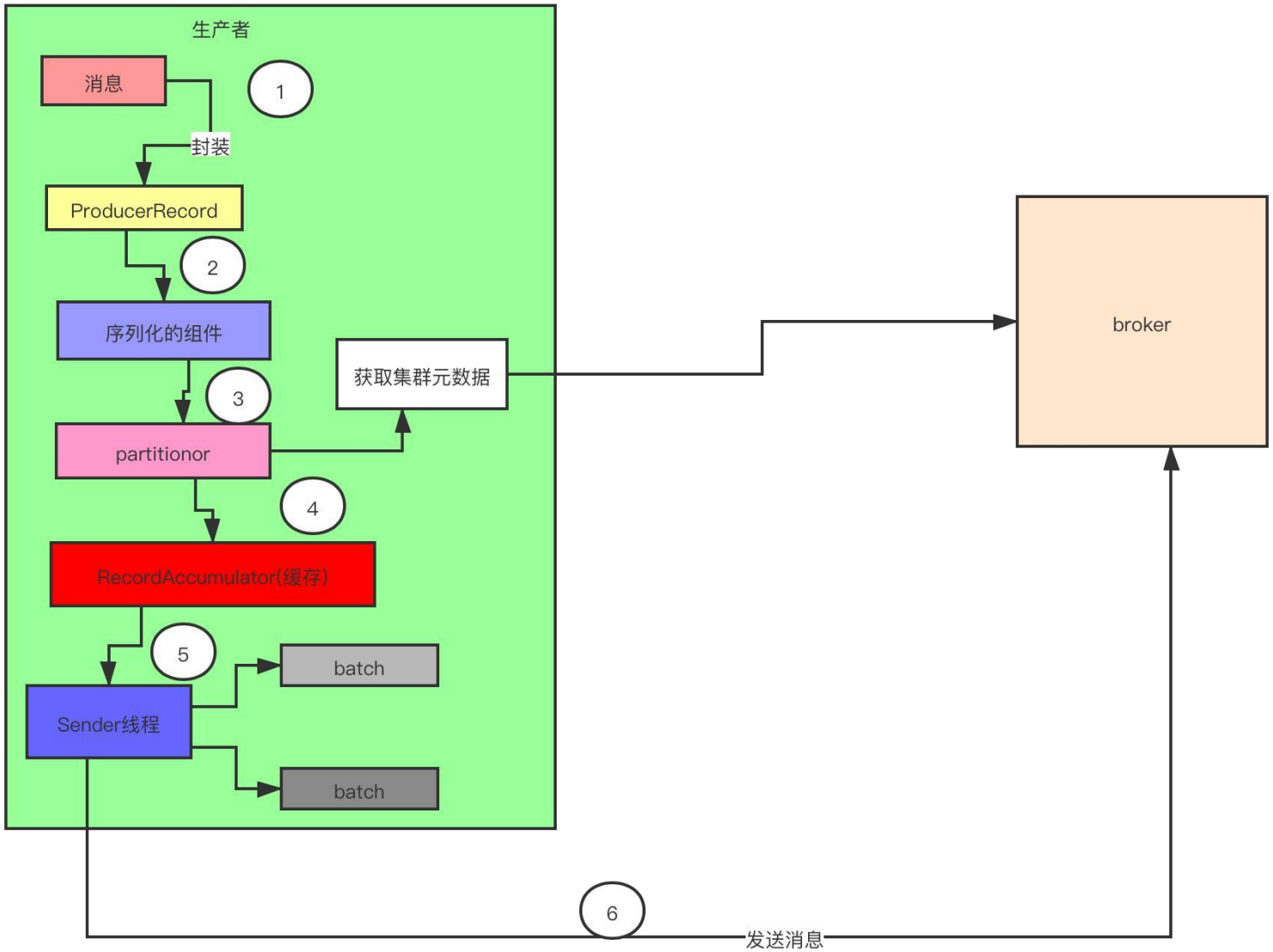
NX 奈学教育

- 我们前面讲的都是服务端的设计，假如是你，你会对Producer端和Consumer做如何设计？

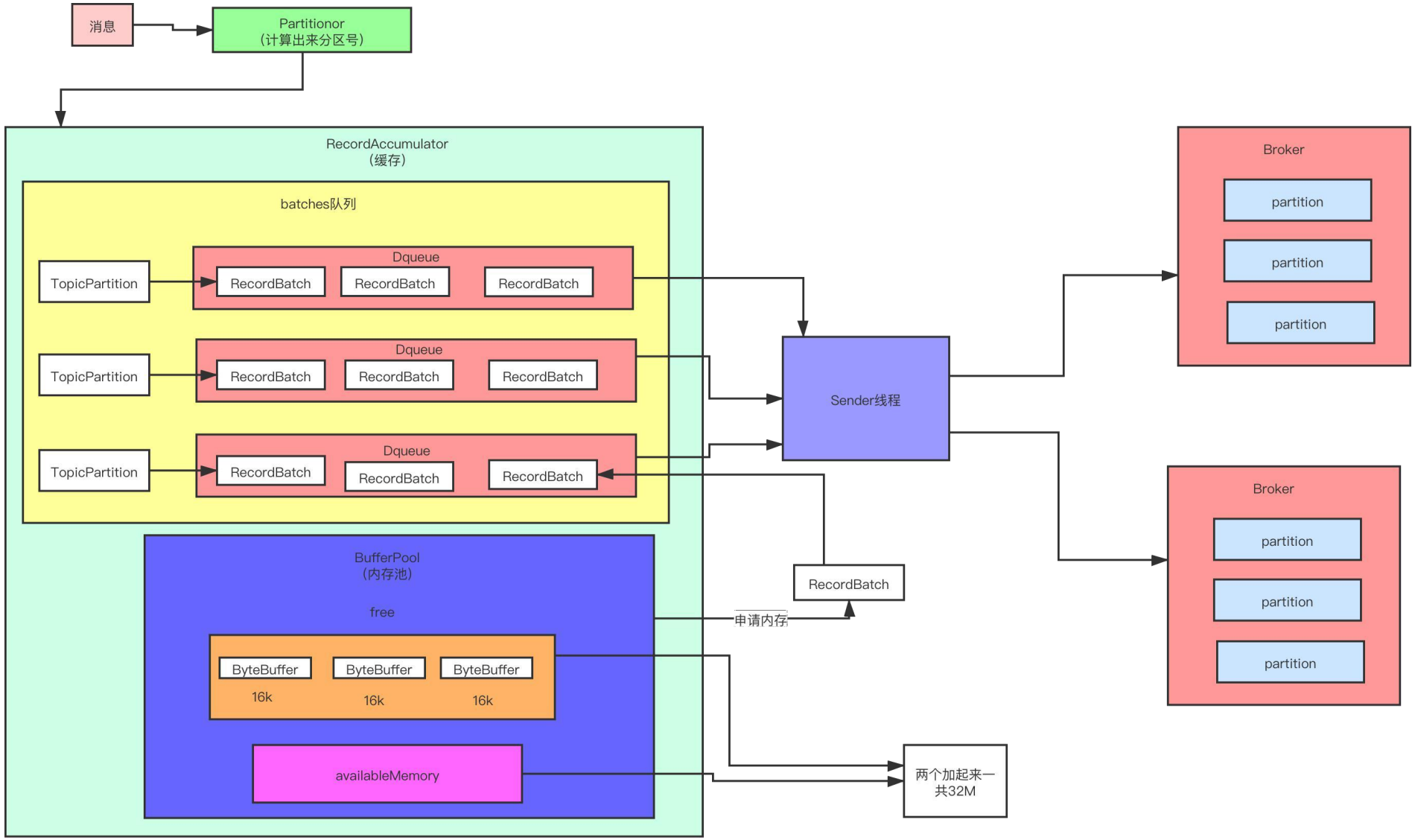


我们先来看看Producer端的设计

Producer设计-批处理



Producer设计-内存池设计



Producer设计-总结

批处理

内存池设计

封装同一服务器请求

看看Consumer端的设计

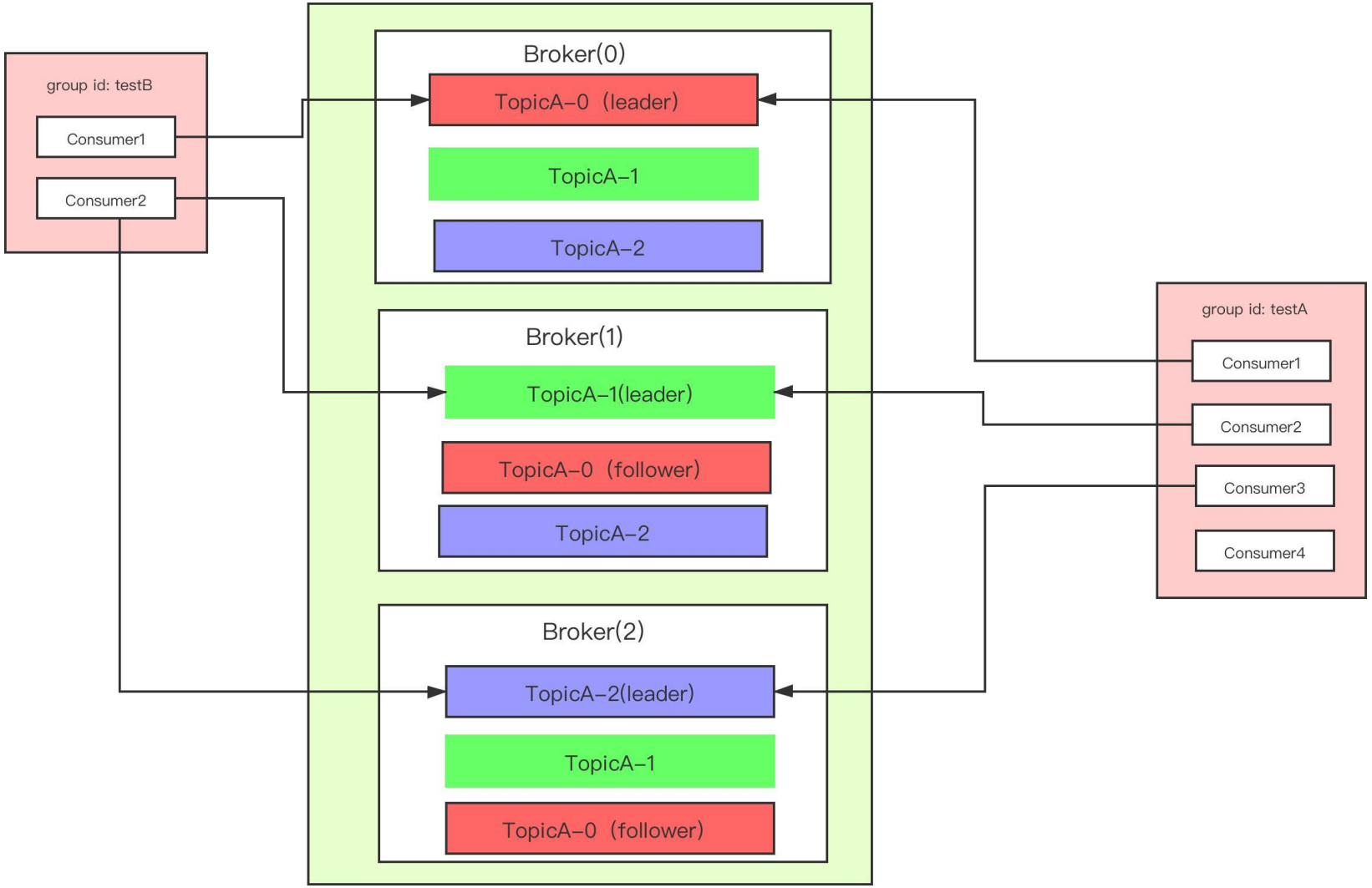
P2P模型和发布订阅模型

NX 奈学教育

P2P模型：也称点对点模型，指同一条消息只能被一个消费者消费，也就是说一个消息如果被这个消费者消费了，其余的消费者就都不能消费了，传统的消息系统用的就是这种方式。

发布订阅模型：允许消息被多个Consumer消费，但是一个Consumer需要订阅主题的所有分区。

Kafka的Consumer Group设计



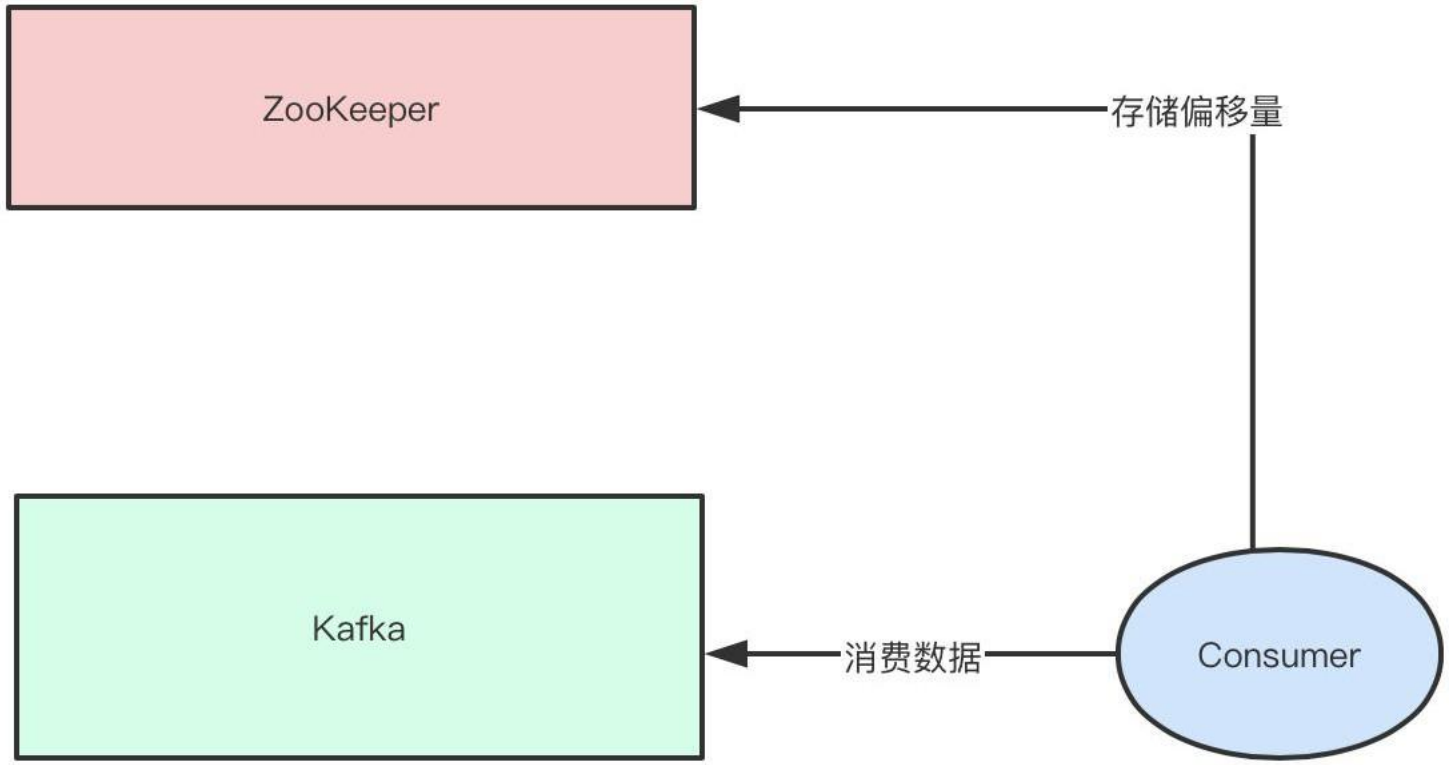
同一个消费组是P2P方式，
一个消息只能被同一个组的一
消费者消费

不同组是订阅模式，
一个消息可以被不同的消费组
消费

一个分区同一时间只会被 同
组一个消费者消费

Consumer设计-偏移量存储

老版本的架构方案（0.8版本）



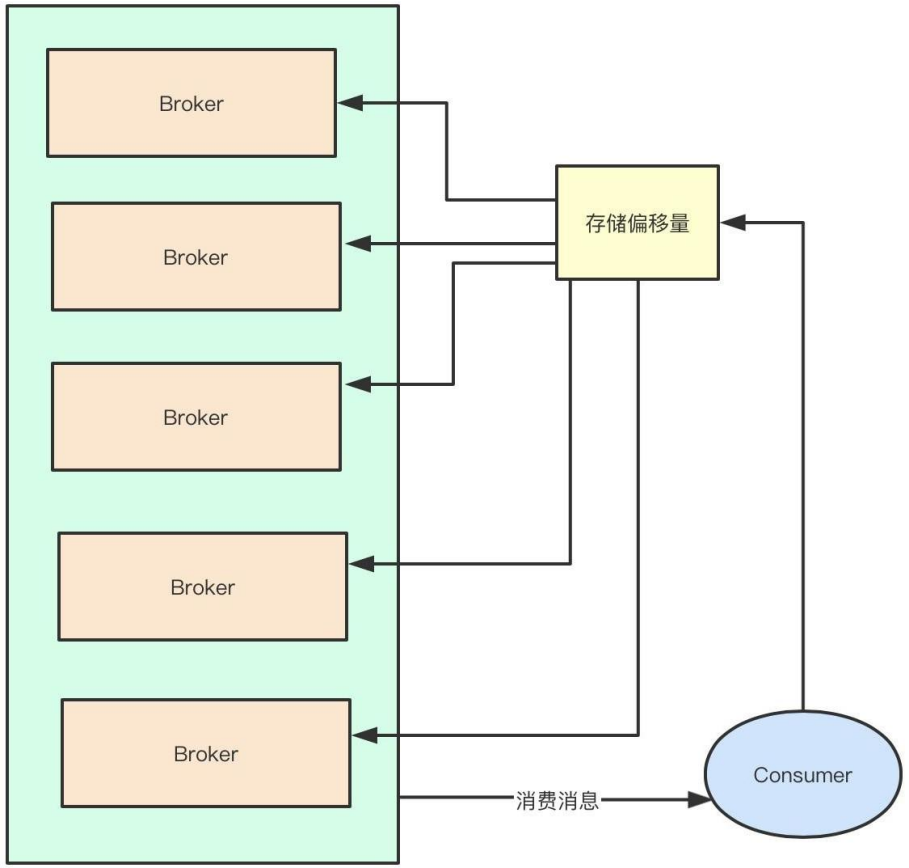
ZK不擅长高并发操作

ZK不适合高频读写操作

Consumer设计-偏移量存储

奈学教育

新版本的架构方案



Topics

Show 10 entries

Topic	# Partitions	# Brokers
__consumer_offsets	50	5

Kafka天然的支持高并发，高可用，高性能

Consumer总结

NX 奈学教育

Consumer Group的设计

偏移量存储改造

课程总结

奈学教育

了解Kafka的特性

掌握Kafka Producer高性能架构设计方案

掌握Kafka服务端高并发，高性能，高可用的架构设计

掌握Kafka提升Consumer的稳定性设计

掌握Kafka的设计，合理对Kafka进行调优

NX 奈学教育

欢迎关注本人公众号
“架构之美”